

PROCESSAMENTO DE IMAGENS EM DISPOSITIVOS MÓVEIS ATRAVÉS DA UTILIZAÇÃO DO MODELO DE CORES HSV

ROBAINA, P. R. ¹, CAMARGO, S. S.¹, MARRANGHELLO F. G. ¹

¹ Universidade Federal do Pampa (UNIPAMPA) – Bagé – RS – Brasil

RESUMO

O presente artigo apresenta o processo de desenvolvimento e análise um aplicativo para processamento de imagens em tempo real para dispositivos mobile. Este trabalho foi desenvolvido no escopo do Projeto Astronomia Para Todos.

Palavras-chave: Android; HSV; Planetário; Astronomia

1 INTRODUÇÃO

A popularização do uso de dispositivos móveis com maior poder de processamento possibilita uma nova gama de aplicações, dentre elas está o processamento de imagens. Esta popularização tem grande potencial de aproveitamento no Projeto Astronomia Para Todos, cujo objetivo principal é difundir conhecimentos sobre Astronomia e estimular professores a estudarem o tema com seus alunos. Neste contexto, o presente trabalho tem como principal objetivo criar uma ferramenta, para dispositivos móveis, que contribua para a disseminação das atividades do Projeto Astronomia para Todos e do Planetário da UNIPAMPA, aproveitando os recursos de dispositivos móveis para compartilhar imagens em redes sociais. Para isso, a ferramenta vai integrar imagens dos visitantes do planetário com imagens de fundo ligadas ao tema Astronomia.

A inspiração para esta proposta advém de alguns trabalhos correlatos. Em [1], a autora utiliza a técnica de visão computacional em um aplicativo para o auxílio de deficientes visuais na travessia de ruas e avenidas. Em [2], os autores propõem uma solução para a identificação de plantas em tempo real, através de uma imagem de suas folhas. Em [3], por sua vez, os autores utilizam o padrão de cores HSV no gerenciamento de carros inteligentes. E, em [4], os autores oferecem uma solução de reconhecimento facial com menor complexidade computacional, utilizando o padrão HSV. Diante de tais exemplos, o presente estudo também apresenta uma aplicação para tratamento de imagens em dispositivos móveis.

O projeto tem como principal objetivo reconhecer um fundo padrão em uma imagem, através da técnica de chroma-key. Assim, remove-se o fundo da imagem original e adiciona-se uma nova imagem de fundo a partir de uma escolha do usuário.

2 METODOLOGIA

Tendo em vista a predominância do sistema Android em smartphones atualmente, o aplicativo foi desenvolvido para esta plataforma. Para o desenvolvimento do aplicativo foi utilizada a IDE (Integrated Development Environment) Android Studio, com linguagem de programação JAVA.

Uma Activity é um módulo único e independente que normalmente está relacionada diretamente com uma tela de interface de usuário e suas funcionalidades correspondentes. Dessa forma, o aplicativo possui duas Activities principais. A Activity principal, Figura 1, é utilizada para obter a imagem a ser processada. A Activity da galeria, Figura 2, apresenta um menu de opções com imagens de fundo. A comunicação, ou transferência de dados entre Activities, é tratada de forma nativa pela plataforma Android.

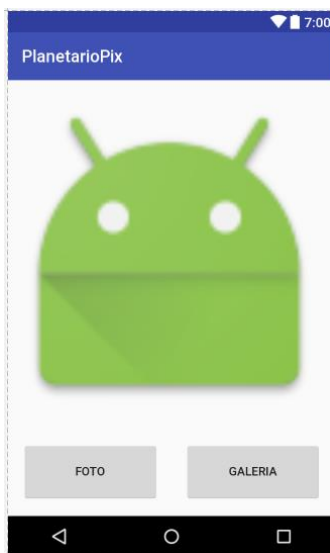


Figura 1. Activity principal

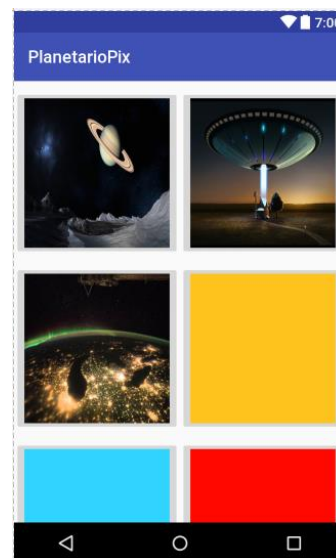


Figura 2. Activity da galeria

Durante o processo, a imagem passa por diversas etapas de processamento, como pode ser observado na Figura 3.

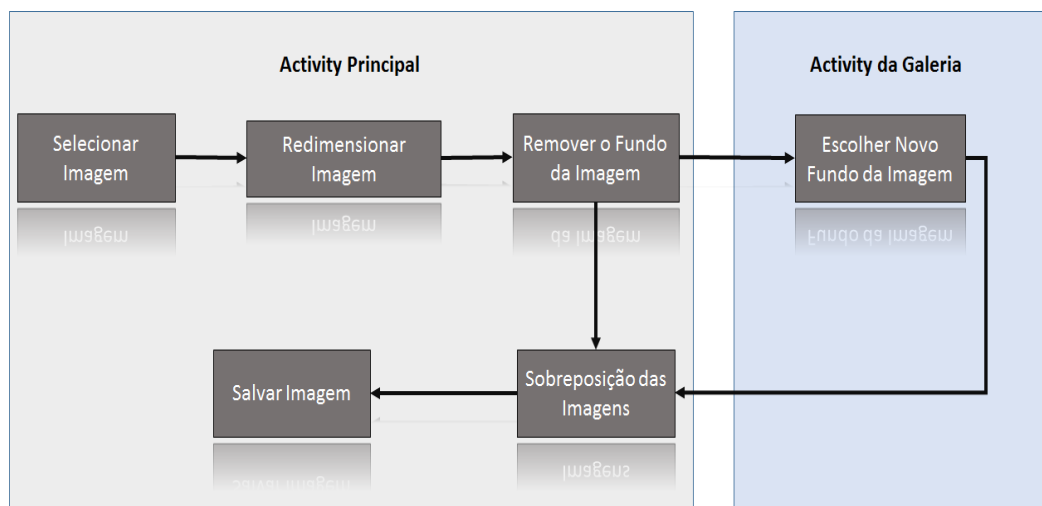


Figura 3. Fluxo do processamento da imagem

Entre as etapas de processamento da imagem estão:

1. **Selecionar Imagem:** O usuário escolhe se irá capturar uma nova imagem através da câmera do dispositivo ou escolher uma imagem da galeria. A imagem capturada ou selecionada é convertida para bitmap.
2. **Redimensionar a Imagem:** Nesta etapa, a imagem é redimensionada para uma resolução de 1280x780 pixels. Como não existe uma padronização de hardware para dispositivos Android, existem dispositivos com resoluções diferentes. Dessa forma, esta etapa é importante pois garante que o aplicativo irá funcionar em qualquer dispositivo.
3. **Remover o Fundo:** Nesta etapa, é utilizado o padrão HSV para a seleção de áreas de interesse, ou seja, o fundo a ser removido. Todos os pixels da imagem são percorridos e os quais são detectadas como fundo são removidas da imagem.
4. **Escolher novo Fundo da Imagem:** Nesta etapa, a imagem selecionada pelo usuário é convertida para bitmap e enviada como retorno para a *Activity* principal. As imagens de fundo são padronizadas na mesma resolução resultante da etapa 2.
5. **Sobreposição das Imagens:** Nesta etapa, um novo bitmap é criado a partir da sobreposição dos insumos da etapa 3 e etapa 4, respectivamente.
6. **Salvar Imagem:** Nesta etapa, o bitmap que contém o resultado final do processo é convertido, ou comprimido no formato JPG, salvo na memória secundária e adicionado a galeria do dispositivo.

O modelo RGB, *red* (vermelho), *green* (verde) e *blue* (azul), é largamente utilizado em dispositivos eletrônicos. Sua representação quantifica quanto cada cor primária (vermelho, verde, azul) faz parte de sua composição.

Em situações reais, variações de luz, sombras ou até deformações no tecido do chroma-key fazem com que mesmo pixels aparentemente iguais possuam valores diferentes uns dos outros, como pode ser visto na Figura 4. Isto implica em o modelo de cores RGB ser ineficaz neste caso, pois a representação de cores RGB não consiste em um bom modelo quando se faz necessário a análise das cores para a seleção de áreas de interesse, dado que não é um sistema onde o espaço de cor é uniforme [1].

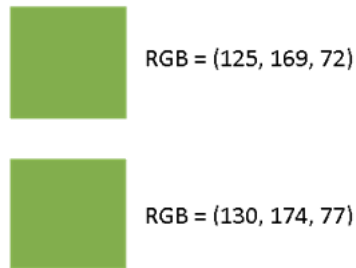


Figura 4. Diferença de valores no modelo RGB

Para aplicações onde as condições de iluminação variadas, existem melhores resultados ao se utilizar modelo de cores que separem os componentes de iluminação matiz e saturação como o HSV, *hue* (matiz), *saturation* (saturação) e *value* (valor). Dessa forma, a etapa 3 apresenta o maior desafio e tempo de processamento do presente projeto. Isso porque as imagens são capturadas em situações reais, com varrições de iluminação.

3 RESULTADOS E DISCUSSÃO

O aplicativo mostrou-se eficaz no processamento de imagens em dispositivos móveis, como pode ser observado na Figura 5. Como o algoritmo leva em consideração a cor de fundo da imagem capturada é necessário que a imagem seja captura com condições de iluminação boas. Para imagens com condições de iluminação não favoráveis, o algoritmo apresentou pequenas falhas na etapa de remoção de fundo.

Visto que nas etapas de remoção de fundo e de sobreposição de imagens a imagem são percorridos todos os pixels da imagem. O tempo de execução do algoritmo apresenta-se insatisfatório.

O tamanho final da aplicação quando instalada no dispositivo é de 10MB, enquanto o tamanho de seu instalador, relevante para futuros *downloads*, é de apenas 2,7MB.

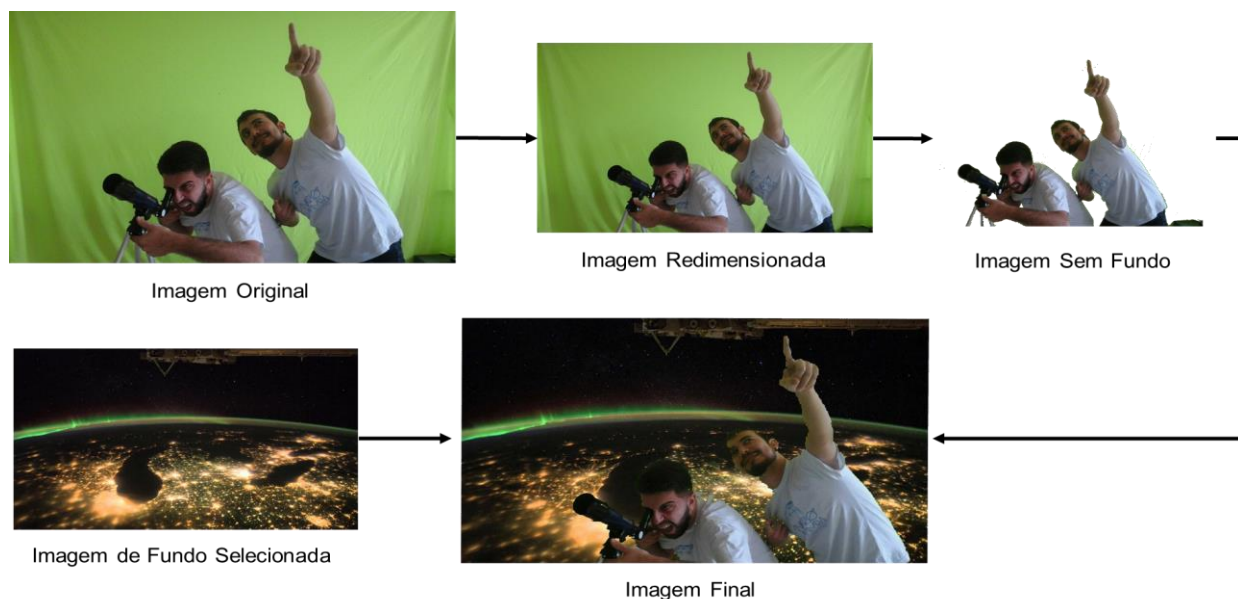


Figura 5. Resultados Práticos Do Aplicativo.

4 CONCLUSÃO

Conforme o esperado, o aplicativo apresentou-se eficaz na solução do problema apontado. Através da utilização do padrão de cores HSV na etapa Remoção de fundo, o aplicativo torna-se funcional, inclusive em situações não ideais de iluminação. O tempo de processamento mostrou-se insatisfatório para os resultados esperados. Em trabalhos futuros, pretende-se aumentar a eficiência do aplicativo buscando formas de diminuir seu tempo de execução e adicionar módulos de compartilhamento das imagens geradas com redes sociais.

5 REFERÊNCIAS

- [1] Sousa, R. (2013). USO DE VISÃO COMPUTACIONAL EM DISPOSITIVOS MÓVIES PARA AUXÍLIO À TRAVESSIA DE PEDESTRES COM DEFICIÊNCIA VISUAL, http://iris.sel.eesc.usp.br/wvc/Anais_WVC2012/pdf/97956.pdf.
- [2] Zhao, Z., Ma, L., Cheung, Y., Wu, X., Tang, Y. & Chen, C. (2015). ApLeaf: An efficient android-based plant leaf identification system, pages 1112–1119. Neurocomputing 151.
- [3] Kwon, Y., Jung, S., Park, D., Sim C. (2013). A Key-Frame Extraction Method based on HSV Color Model for Smart Vehicle Management System, pages 595-604, http://www.riss.kr/search/detail/DetailView.do?p_mat_type=1a0202e37d52c72d&control_no=0a93d6eeb20617c7ffe0bdc3ef48d419#redirect
- [4] Li, X., Xu, T. (2011). Face Video Key-Frame Extraction Algorithm Based on Color Histogram, pages 670-676. International Conference on Computer Sciences and Information Technology 2011.